UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/756,986	01/13/2004	Shirish Hemant Phatak	102916-001 US	3709

23526 7590 11/08/2005

NORRIS MCLAUGHLIN & MARCUS, P.A.
P O BOX 1018
SOMERVILLE, NJ 08876

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 11/08/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/756,986	PHATAK, SHIRISH HEMANT	
	Examiner	Art Unit	
	Luke S. Wassum	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 August 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 January 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. The Applicant's amendment, filed 29 August 2005, has been received, entered into the record, and considered.

2. As a result of the amendment, claims 1 and 15 have been amended. Claims 1-22 remain pending in the application.

Priority

3. The Applicant's claim to domestic priority under 35 U.S.C. § 119(e) to U.S. Provisional Application 60/440,750, filed 17 January 2003, is acknowledged.

4. Since the subject matter and disclosure of the priority document substantially encompasses that of the instant application, a priority date of 17 January 2003 is established.

The Invention

5. The claimed invention is for a distributed caching method wherein data file coherency and consistency are maintained through the use of storage caches associated with a plurality of client/remote workstations, and a cache server associated with a file server.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

8. Claims 11-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Barbará et al.** (U.S. Patent 5,706,435) in view of **McClain et al.** (U.S. Patent 6,049,874) in view of **Draper et al.** (U.S. Patent 5,924,096) in view of **Battersby et al.** (U.S. Patent 5,740,370).

9. Regarding claim 11, **Barbará et al.** teaches a method for managing shared access to data files stored in a file server by a plurality of authorized workstations substantially as claimed, the method comprising:

a) automatically transmitting file update data from a cache server to a first storage cache (see disclosure of the transmission of invalidity reports from a server cache to a client

cache, and the subsequent transfer of the current value of any datum that is needed and not resident in the client cache, col. 4, lines 7-37; see also col. 6, lines 46-56); and

b) incorporating the file update data into the cached data file at the first storage cache such that the cached data file is updated to be the same as the data file currently stored at the file server or the cache server (see disclosure of the transmission of invalidity reports from a server cache to a client cache, and the subsequent transfer of the current value of any datum that is needed and not resident in the client cache, col. 4, lines 7-37; see also col. 6, lines 46-56).

Barbará et al. does not explicitly teach a method wherein the file update data is a function of differences between the data file as currently stored at the file server or the cache server and a cached data file stored at the first storage cache and corresponding to the data file.

McClain et al., however, teaches a method wherein the file update data is a function of differences between the data file as currently stored at the file server or the cache server and a cached data file stored at the first storage cache and corresponding to the data file (see disclosure of a system that backs up files as changes occur, by transmitting only the changed portions of the files, Abstract; see also col. 2, lines 61-64; see also col. 4, lines 44-45).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit only those portions of files that have changed, since it is advantageous to transfer as little data as possible while supporting the goal of transferring changes to the files, because this would preserve bandwidth on the network.

Neither **Barbará et al.** nor **McClain et al.** explicitly teaches a method wherein said automatic transmitting is performed in response to a workstation request for access to a file.

Draper et al., however, teaches a method wherein a cache us updated in response to a request to access the cache (see col. 7, lines 60-61).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate a cache updating policy as disclosed, since this would ensure that the user always accessed the latest version of a requested file, while at the same time not requiring the added processing and bandwidth burden of updating all files in all caches immediately upon updating the file on the server.

None of **Barbará et al.**, **McClain et al.** nor **Draper et al.** explicitly teaches a method wherein the server cache is separate from the file server.

Battersby et al., however, teaches a method wherein the server cache resides on a shared cache server, separate from the file server (see Abstract; see also file server 22 and cache server 20 in Figure 1; see also col. 2, lines 56-65).

It would have been obvious to one of ordinary skill in the art at the time of the invention to include a cache server separate from the file server, since this would reduce the burden on the

network and network servers, and provide the connected client computers with faster access to frequently used data (see col. 2, lines 48-51).

10. Regarding claim 12, **McClain et al.** additionally teaches a method wherein the file update data is transmitted as streaming data to the first storage cache (see disclosure of the analogous teaching that changes are transmitted as the user makes them, col. 4, line 60 through col. 5, line 5; see also disclosure that the backing up can be performed continuously, col. 2, lines 55-58).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit the file update data as streaming data, since this would ensure timely update of the file on the server cache and the file server.

11. Regarding claim 13, **McClain et al.** additionally teaches a method further comprising compressing the file update data prior to transmission to the first storage cache (see col. 9, lines 31-36).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ compression when transferring file changes between the file server, server cache and storage caches, since compression minimizes the amount of data that needs to be transmitted, thus preserving bandwidth on the networks.

12. Regarding claim 14, **Barbará et al.** additionally teaches a method wherein the automatically transmitting and the incorporating steps are performed at predetermined intervals (see disclosure that invalidity reports are transmitted periodically, col. 4, lines 7-24; see also col. 5, lines 37-40).

13. Claims 1-7 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Stiles** (U.S. Patent 5,594,863) in view of **McClain et al.** (U.S. Patent 6,049,874) in view of **Battersby et al.** (U.S. Patent 5,740,370).

14. Regarding claim 1, **Stiles** teaches a method for managing shared access to data files stored in a file server by a plurality of authorized computer workstations as claimed, the method comprising:

a) supplying to a first storage cache a copy of a data file retrieved from the file server by a cache server for reading or updating, wherein the first storage cache is associated with at least a first authorized computer workstation and stores the copy of the data file as a cached data file (see disclosure of server cache and client cache, col. 2, lines 18-38);

b) at the first storage cache, incorporating data file modifications entered by the first workstation into the cached data file as the modifications are being entered, such that the cached data file is a current version (see col. 2, lines 18-38; see also col. 2, line 66 through col. 3, line 8; see also col. 3, lines 30-39);

c) automatically transmitting file update data from the first storage cache to the cache server which make the cached data file the current version (see col. 2, line 66 through col. 3, line 8; see also col. 3, lines 30-39); and

d) at the cache server, generating a replacement version of the data file stored at the file server based on the file update data (see col. 2, line 66 through col. 3, line 8; see also col. 3, lines 30-39).

Stiles does not explicitly teach a method wherein the file update data is a function of the modifications incorporated into the cached data file.

McClain et al., however, teaches a method wherein the file update data is a function of the modifications incorporated into the cached data file (see disclosure of a system that backs up files as changes occur, by transmitting only the changed portions of the files, Abstract; see also col. 2, lines 61-64; see also col. 4, lines 44-45).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit only those portions of files that have changed, since it is advantageous to transfer as little data as possible while supporting the goal of transferring changes to the files, because this would preserve bandwidth on the network.

Neither **Stiles** nor **McClain et al.** explicitly teaches a method wherein the server cache is separate from the file server.

Battersby et al., however, teaches a method wherein the server cache resides on a shared cache server, separate from the file server (see Abstract; see also file server 22 and cache server 20 in Figure 1; see also col. 2, lines 56-65).

It would have been obvious to one of ordinary skill in the art at the time of the invention to include a cache server separate from the file server, since this would reduce the burden on the network and network servers, and provide the connected client computers with faster access to frequently used data (see col. 2, lines 48-51).

15. Regarding claim 15, **Stiles** teaches a system for managing shared access to data files stored in a file server by a plurality of authorized computer workstations substantially as claimed, the system comprising:

a) a cache server for retrieving a copy of a data file stored at the file server (see discussion of server cache at col. 2, lines 18-30);

b) at least a first storage cache associated with at least a first authorized computer workstation, wherein the first storage cache stores in a local memory the retrieved data file copy as a cached data file and incorporates data file modifications entered by the first workstation into the cached data file as the modifications are being entered, such that the cached data file is a current version (see discussion of the client cache, col. 2, lines 18-38; see also col. 2, line 66 through col. 3, line 8; see also col. 3, lines 30-39);

c) wherein the first storage cache automatically generates and transmits file update data to the cache server which makes the cached data file the current version (see col. 2, line 66 through col. 3, line 8; see also col. 3, lines 30-39); and

d) wherein the cache server generates a replacement version of the data file based on the file update data and responds to a request for access to the data file subsequently

transmitted to the cache server from at least one of a second storage cache and an authorized computer workstation using the replacement version of the data file (see disclosure that the server cache provides clients with the most up to date version of the data file, col. 2, lines 18-30).

Stiles does not explicitly teach a system wherein the file update data is a function of the modifications incorporated into the cached data file, nor wherein file update data is transmitted to the cache server as streaming data.

McClain et al., however, teaches a system wherein the file update data is a function of the modifications incorporated into the cached data file (see disclosure of a system that backs up files as changes occur, by transmitting only the changed portions of the files, Abstract; see also col. 2, lines 61-64; see also col. 4, lines 44-45).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit only those portions of files that have changed, since it is advantageous to transfer as little data as possible while supporting the goal of transferring changes to the files, because this would preserve bandwidth on the network.

McClain et al. additionally teaches a method wherein the file update data is transmitted as streaming data to the first storage cache (see disclosure of the analogous teaching that changes are transmitted as the user makes them, col. 4, line 60 through col. 5, line 5; see also disclosure that the backing up can be performed continuously, col. 2, lines 55-58).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit the file update data as streaming data, since this would ensure timely update of the file on the server cache and the file server.

Neither **Stiles** nor **McClain et al.** explicitly teaches a method wherein the server cache is separate from the file server.

Battersby et al., however, teaches a method wherein the server cache resides on a shared cache server, separate from the file server (see Abstract; see also file server 22 and cache server 20 in Figure 1; see also col. 2, lines 56-65).

It would have been obvious to one of ordinary skill in the art at the time of the invention to include a cache server separate from the file server, since this would reduce the burden on the network and network servers, and provide the connected client computers with faster access to frequently used data (see col. 2, lines 48-51).

16. Regarding claim 2, **Stiles** additionally teaches a method for managing shared access to data files further comprising at the cache server, if the file is accessed for updating by the first storage cache, protecting the data file stored at the file server from updates from other storage caches until all file update data from the first storage cache has been incorporated into the replacement version of the data file and the replacement version has replaced the data file stored at the file server (see

disclosure of the use of file locks to ensure consistency of file versions across the distributed system, col. 2, lines 50-65).

17. Regarding claim 3, **Stiles** additionally teaches a method for managing shared access to data files wherein the protecting comprises protecting the data file stored at the file server from updates from other storage caches while the file update data from the first storage cache is transmitted to the cache server (see disclosure of the use of file locks or a "write-through" technique to ensure consistency of file versions across the distributed system, col. 2, line 50 through col. 3, line 8).

18. Regarding claim 4, **Stiles** additionally teaches a method for managing shared access to data files further comprising:

a) replacing the data file stored at the file server with the replacement version of the data file (see col. 2, lines 18-30; see also col. 2, line 50 through col. 3, line 8); and

b) responding to a request for access to the data file subsequently transmitted to the cache server from at least one of a second storage cache and an authorized computer workstation using the replacement version of the data file (see col. 2, lines 18-30).

19. Regarding claim 5, **McClain et al.** additionally teaches a method wherein the file update data is transmitted as streaming data to the first storage cache (see disclosure of the analogous teaching that changes are transmitted as the user makes them, col. 4, line 60 through col. 5, line 5; see also disclosure that the backing up can be performed continuously, col. 2, lines 55-58).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit the file update data as streaming data, since this would ensure timely update of the file on the server cache and the file server.

20. Regarding claim 6, **McClain et al.** additionally teaches a method further comprising compressing the file update data prior to transmission to the first storage cache (see col. 9, lines 31-36).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ compression when transferring file changes between the file server, server cache and storage caches, since compression minimizes the amount of data that needs to be transmitted, thus preserving bandwidth on the networks.

Regarding claim 7, **McClain et al.** additionally teaches a method wherein the file update data includes difference data, wherein the difference data represents the difference between the cache data file at the first storage cache and the version of the data file currently stored at the file server or the cache server (see disclosure of a system that backs up files as changes occur, by transmitting only the changed portions of the files, Abstract; see also col. 2, lines 61-64; see also col. 4, lines 44-45).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit only those portions of files that have changed, since it is advantageous to transfer as little data as possible while supporting the goal of transferring changes to the files, because this would preserve bandwidth on the network.

21. Claims 10 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Stiles** (U.S. Patent 5,594,863) in view of **McClain et al.** (U.S. Patent 6,049,874) in view of **Battersby et al.** (U.S. Patent 5,740,370) as applied to claims 1-7 and 15 above, and further in view of **Barbará et al.** (U.S. Patent 5,706,435).

22. Regarding claim 10, **Stiles, McClain et al.** and **Battersby et al.** teach a method for managing shared access to data files substantially as claimed.

None of **Stiles, McClain et al.** nor **Battersby et al.** explicitly teaches a method wherein the file update data is automatically transmitted to the cache server at predetermined intervals.

Barbará et al., however, teaches a method wherein the automatically transmitting step is performed at predetermined intervals (see analogous disclosure that invalidity reports are transmitted periodically, col. 4, lines 7-24; see also col. 5, lines 37-40).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit file update data on a periodic basis, since this would server to transfer multiple distinct file changes to the cache server in a single transmission (as opposed to transmitting each distinct file change in a separate transmission), thus preserving bandwidth on the network.

23. Regarding claim 16, **Stiles, McClain et al.** and **Battersby et al.** teach a system for managing shared access to data files substantially as claimed.

None of **Stiles, McClain et al.** nor **Battersby et al.** explicitly teaches a system wherein the cache server generates server update file data and automatically transmits the server update file data to the first storage cache.

Barbará et al., however, teaches a system including automatically transmitting file update data from a cache server to a first storage cache in response to a workstation request for access to a data file which is stored at a file server associated with the cache server (see disclosure of the transmission of invalidity reports from a server cache to a client cache, and the subsequent transfer of the current value of any datum that is needed and not resident in the client cache, col. 4, lines 7-37; see also col. 6, lines 46-56).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the claimed features, since this is an efficient mechanism for ensuring that data stored in the first storage cache is not stale, thus maintaining a coherent view of the data across the storage caches.

24. Claims 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Stiles** (U.S. Patent 5,594,863) in view of **Rao et al.** (U.S. Patent 5,689,706) in view of **Battersby et al.** (U.S. Patent 5,740,370).

25. Regarding claim 17, **Stiles** teaches a system for managing shared access to data files stored in a file server by a plurality of authorized computer workstations substantially as claimed, the system comprising:

a) a cache server for coupling to the file server (see discussion of server cache at col. 2, lines 18-30);

b) a plurality of storage caches for accessing data files stored in the file server by establishing a communications connection with the cache server (see discussion of the client cache, col. 2, lines 32-38; see also col. 2, line 66 through col. 3, line 8; see also col. 3, lines 30-39); and

c) wherein the cache server automatically performs steps to update the cached data file at the first storage cache (see disclosure that the server cache provides clients with the most up to date version of the data file, col. 2, lines 18-30).

Stiles does not explicitly teach a system including a leasing module implementing rules for deciding whether to grant or deny a request for a lease for a data file based on the presence and/or type of existing lease for the data file, wherein the decision is made in accordance with the criteria that a write lease cannot be granted if a read lease already exists, only a reader right can be granted if a write lease already exists and an additional read lease can be granted if a read lease already exists.

Rao et al., however, teaches a system including an analogous token server which regulates the reading and writing of files based on read and write tokens (analogous to the claimed leases), including conventional rules governing the granting and denying of read and write tokens to ensure

the integrity of each version of each data file (see extensive discussion at col. 15, line 62 through col. 19, line 12, and particularly col. 15, lien 62 through col. 16, line 53).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate a mechanism to govern the reading and writing of data files amongst the server and various workstations, since without such a mechanism there would be no way to guarantee that multiple workstations did not attempt to update a particular data file at the same time, resulting in the loss of some file update data.

Neither **Stiles** nor **Rao et al.** explicitly teaches a system wherein the server cache is separate from the file server.

Battersby et al., however, teaches a system wherein the server cache resides on a shared cache server, separate from the file server (see Abstract; see also file server 22 and cache server 20 in Figure 1; see also col. 2, lines 56-65).

It would have been obvious to one of ordinary skill in the art at the time of the invention to include a cache server separate from the file server, since this would reduce the burden on the network and network servers, and provide the connected client computers with faster access to frequently used data (see col. 2, lines 48-51).

26. Regarding claim 18, **Stiles** teaches a system for managing shared access to data files stored in a file server by a plurality of authorized computer workstations substantially as claimed, the system comprising:

a) a cache server for coupling to the file server (see discussion of server cache at col. 2, lines 18-30);

b) a plurality of storage caches for accessing data files stored in the file server by establishing a communications connection with the cache server (see discussion of the client cache, col. 2, lines 32-38; see also col. 2, line 66 through col. 3, line 8; see also col. 3, lines 30-39); and

c) wherein the cache server automatically performs steps to update the cached data file at the first storage cache (see disclosure that the server cache provides clients with the most up to date version of the data file, col. 2, lines 18-30).

Stiles does not explicitly teach a system wherein each storage cache includes a leasing module implementing rules for deciding whether to grant or deny a request for a lease for a data file based on the presence and/or type of existing lease for the data file, wherein the decision is made in accordance with the criteria that a write lease cannot be granted if a read lease already exists, only a reader right can be granted if a write lease already exists and an additional read lease can be granted if a read lease already exists.

Rao et al., however, teaches a system wherein each storage cache includes an analogous token server which regulates the reading and writing of files based on read and write tokens (analogous to the claimed leases), including conventional rules governing the granting and denying

of read and write tokens to ensure the integrity of each version of each data file, including the steps of determining the lease condition for the data file at the storage cache, granting the request under certain conditions, requesting a new lease under certain conditions, and denying the request under certain conditions (see disclosure that there is a token server on each component of the system, col. 3, lines 1-17; see also extensive discussion at col. 15, line 62 through col. 19, line 12, and particularly col. 15, lien 62 through col. 16, line 53; see also Figure 14).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate a mechanism to govern the reading and writing of data files amongst the server and various workstations, since without such a mechanism there would be no way to guarantee that multiple workstations did not attempt to update a particular data file at the same time, resulting in the loss of some file update data.

Neither **Stiles** nor **Rao et al.** explicitly teaches a system wherein the server cache is separate from the file server.

Battersby et al., however, teaches a system wherein the server cache resides on a shared cache server, separate from the file server (see Abstract; see also file server 22 and cache server 20 in Figure 1; see also col. 2, lines 56-65).

It would have been obvious to one of ordinary skill in the art at the time of the invention to include a cache server separate from the file server, since this would reduce the burden on the

network and network servers, and provide the connected client computers with faster access to frequently used data (see col. 2, lines 48-51).

27. Regarding claim 19, **Stiles** additionally teaches a system for managing shared access to data files wherein the automatic updating steps include updating the data file stored at the file server based on a cached data file stored at one of the storage caches (see col. 2, lines 18-37).

28. Regarding claim 20, **Rao et al.** additionally teaches a system wherein the leasing module at a first storage cache performs the following steps when the leasing module determines that a cached data file corresponding to a data file stored at the file server is no longer opened at a workstation associated with the first at storage caches:

a) determining whether the cached data file copy was modified based on entries made by the workstation while the cached data file was open (see col. 17, line 49 through col. 19, line 12, particularly col. 18, lines 15-26);

b) if the data file copy was modified, automatically transmitting file update data to the cache server (see col. 17, line 49 through col. 19, line 12, particularly col. 18, lines 15-26); and

c) releasing any lease for the data file (see col. 17, line 49 through col. 19, line 12, particularly col. 18, lines 15-26);

wherein the cache server uses the file update data to update the version of the data file stored on the file server.

29. Claims 21 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Stiles** (U.S. Patent 5,594,863) in view of **Rao et al.** (U.S. Patent 5,689,706) in view of **Battersby et al.** (U.S. Patent 5,740,370) as applied to claims 17-20 above, and further in view of **McClain et al.** (U.S. Patent 6,049,874).

30. Regarding claims 21 and 22, **Stiles**, **Rao et al.** and **Battersby et al.** teach a system for managing shared access to data files substantially as claimed, but fail to explicitly teach a system including file update data that is transmitted as compressed or streaming data.

31. Regarding claim 21, **McClain et al.** additionally teaches a system wherein the file update data is transmitted as streaming data to the first storage cache (see disclosure of the analogous teaching that changes are transmitted as the user makes them, col. 4, line 60 through col. 5, line 5; see also disclosure that the backing up can be performed continuously, col. 2, lines 55-58).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transmit the file update data as streaming data, since this would ensure timely update of the file on the server cache and the file server.

32. Regarding claims 21 and 22, **McClain et al.** additionally teaches a system further comprising compressing the file update data prior to transmission to the first storage cache (see col. 9, lines 31-36).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ compression when transferring file changes between the file server, server cache and storage caches, since compression minimizes the amount of data that needs to be transmitted, thus preserving bandwidth on the networks.

33. Claims 8 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Stiles** (U.S. Patent 5,594,863) in view of **McClain et al.** (U.S. Patent 6,049,874) in view of **Battersby et al.** (U.S. Patent 5,740,370) as applied to claims 1-7 and 15 above, and further in view of **Bereznyi et al.** (U.S. Patent 6,453,404).

34. Regarding claims 8 and 9, **Stiles**, **McClain et al.** and **Battersby et al.** teach a method for managing shared access to data files substantially as claimed, but fail to explicitly teach a method including a plurality of cache servers.

Bereznyi et al., however, teaches a method including distributed cache providing fault-tolerant operation through the use of redundant cache servers (see disclosure of the primary cache and the failback cache, Figure 15; see also col. 38, line 36 through col. 39, line 42).

It would have been obvious to one of ordinary skill in the art at the time of the invention to include redundant caching, since this would provide more fault-tolerant operation of the system.

Response to Arguments

35. Applicant's arguments filed 29 August 2005 have been fully considered but they are not persuasive.

36. Regarding the Applicant's argument that the newly added limitation that the cached version is updated such that the cached data file is the current version distinguishes over the prior art, the examiner respectfully disagrees.

As cited in the rejection of record, the **Stiles** reference discloses in col. 2, line 66 et seq. that one method for addressing the problem of data inconsistencies between various client caches and the server cache is to force all write requests all the way to the server store each time they are made. Such a system anticipates the claimed "automatically transmitting file update data from the first storage cache (the client cache) to the cache server (the server cache) which makes the cached data file the current version".

37. Regarding the Applicant's argument that the **Draper et al.** reference fails to teach a cache server and first storage cache interfacing with a file server and a plurality of workstations, the examiner respectfully responds that these features are found in the **Barbará et al.** reference, as stated in the rejection of record. The **Draper et al.** reference is relied upon merely for the feature of cache updating as a result of a request for access to a file.

One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

38. Regarding the Applicant's argument that the **Rao et al.** reference fails to teach a data file sharing system including a plurality of storage caches and a cache server, the examiner respectfully responds that these features are found in the **Stiles** reference, as stated in the rejection of record. The **Rao et al.** reference is relied upon merely for the features of the claimed leasing module.

One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

39. Regarding the Applicant's argument that the **Rao et al.** reference fails to teach leasing provisions which grants reader rights if a write lease already exists, the examiner respectfully responds that the reference teaches three levels of privilege; write, sequential read and atomic read. At col. 16, lines 1-3, the reference teaches that atomic read privileges are granted even if a write token exists.

Conclusion

40. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be

calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Luke S. Wassum whose telephone number is 571-272-4119. The examiner can normally be reached on Monday-Friday 8:30-5:30, alternate Fridays off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene can be reached on 571-272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

In addition, INFORMAL or DRAFT communications may be faxed directly to the examiner at 571-273-4119. Such communications must be clearly marked as INFORMAL, DRAFT or UNOFFICIAL.

Customer Service for Tech Center 2100 can be reached during regular business hours at (571) 272-2100, or fax (571) 273-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Luke S. Wassum
Primary Examiner
Art Unit 2167

lsw
7 November 2005